

### 3. Diego

**Program Name: Diego.java**

**Input File: diego.dat**

Diego has found a fancy digital safe but is having trouble opening it. The combination for the safe is a series of digits (0-9). Each time he tries a combination, the safe gives him two pieces of information:

- 1) The number of digits that belong in the correct combination from those that he picked.
- 2) The number of digits that are in the correct position from those that he picked.

Note that excess digits are not counted in total (1). For instance, if the code was 123, and Diego guessed 411, the safe's response would be 1 0 rather than 2 0. Given a set of Diego's guesses and the safe's responses, write a program to help Diego determine how many combinations could possibly open the safe.

**Input:**

The first line of input will contain a single integer T, the number of test cases to follow ( $1 \leq T \leq 10$ )

The first line of each test case will contain two space separated integers N and M denoting the length of the safe's code ( $1 \leq N \leq 6$ ) and the number of observations ( $1 \leq M \leq 10$ )

The next M lines of each test case will each consist of a single observation of the form G X Y, where G is a guess consisting of N digits (0-9), and X and Y are integers denoting how many digits belong in the code, and how many digits are in the correct position.

**Output:**

For each test case on its own line, output the number of safe combinations that are consistent with Diego's observations.

**Sample input:**

```
2
2 4
02 1 0
34 0 0
56 0 0
78 0 0
5 2
12345 0 0
67890 0 0
```

**Sample output:**

```
4
0
```

**Sample Explanation:**

Combinations that are consistent with test case 1: 10, 90, 21, 29

There are no combinations consistent with test case 2, because every digit was used and the safe reported that none of them are present in the combination